I'm human

How to generate report in cucumber

How to generate extent report in cucumber junit. How to generate report in cucumber framework. How to generate extent report in cucumber testng. How to generate extent report in cucumber. How to generate pdf report in cucumber. How to generate testng report in cucumber. How to generate cucumber report in intellij. How to generate allure report in cucumber. How to generate allure report in selenium cucumber. How to generate cucumber report in cypress. How to generate allure report in selenium cucumber java. How to generate extent pdf report in cucumber. How to generate serenity report in cucumber. How to generate cucumber report in karate framework.

Cucumber integrates reporter plugins to generate reports showcasing passed or failed scenarios. Built-in and custom formatters are available, along with third-party plugins for various programming languages. The Cucumber Reports service offers an easy entry point, supported by Cucumber-JVM 6.7.0+, Cucumber-Ruby 5.1.1+, and Cucumber-JS 7.0.0+. If you prefer local reports, use built-in formatters like message, progress, pretty, html, json, rerun, junit, or testng. You can also extend Cucumber by creating your own formatter class implementing the standard interface. Add your custom formatter to the features/support directory and reference it with the --format flag: cucumber --format MyModule::CustomFormatter. Cucumber's formatter API uses an event-based approach, responding to defined events under the Cucumber Messages standard. Third-party plugins include Masterthought TeamCity for integrating results in a format compatible with a TeamCity build agent, TextmateFormatter for printing HTML reports with enhanced styling and JavaScript, and more. CucumberJS Test Reporting Integration for ReportPortal Dashboard ========== Bilgerat: A HipChat Room-Based Failure Message Formatter ---------- * Sends failure messages to designated HipChat rooms for real-time reporting. cucumber_statistics: Timing and Result Tracking in a Single HTML Page ---------- * Tracks timing and displays results in a single HTML page. * Highlights outliers in a table sortable by various metrics. cucumber_characteristics: HTML/JSON Reports on Test Timings and Usage ---------- * Generates HTML/JSON reports on overall test timings, as well as timings and usage of Steps, Features, and Examples. * Lists unused and ambiguous (Cucumber 1.x) Steps. * Compatible with Cucumber 1.x and 2.1+, and Ruby 1.9+. Allure-Cucumber: Allure Adaptor for Cucumber Reporting ---------- * This formatter generates XML files for Allure reporting framework. Cluecumber: A Maven Plugin for Clear and Concise Cucumber Reporting ---------- * A Maven plugin for clear and concise Cucumber reporting. Cucelastic: A Maven Plugin to Push Test Report Data into Elastic Search ---------- * A Maven plugin to push test report data into Elastic Search. * Enables users to visualize a dynamic and easy sharable report/dashboard with the possibility to filter and analyze the data, extend and share it across teams. Serenity/JS: An Acceptance Testing and Reporting Framework ---------- * An acceptance testing and reporting framework with in-depth HTML reports. * Screenplay Pattern APIs, and support for every single version of Cucumber.js Lippia Test Manager: Combining Manual and Automated Test Runs using Gherkin ---------- * Combines the use of Gherkin to support manual and automated test runs using Cucumber. * Allows users to write tests in the Gherkin language without needing an integrated development environment (IDE) and to store them in code repositories. * Eliminates the need for mapping between manual and automated scenarios since it utilizes the .feature files. Pretty Format: Generating HTML Reports in Cucumber ---------- * Generates the Cucumber test report in the HTML format, i.e., an HTML file. * The most readable report format. * Generates the report in the same way as it is a feature file so it becomes easy to track the test results. * Can specify the location where we want this report to be placed after the test execution. Cucumber Reporting Plugin: A Cucumber Plugin for Pretty HTML Reports ---------- * A Cucumber plugin which produces pretty HTML reports using cucumber-reporting. cucumber-java 7.2.3 junit 4.13.2 io.cucumber cucumber-junit 7.2.3 Step 3: Create a package named "Feature" under src/test/resources. Step 4: Create CucumberReport.feature as the feature file. Feature: Check for Login functionality Scenario: Welcome Page Login verification Given User is on Welcome Page When User clicks on Login button Then Welcome page should be displayed Step 5: Create a package for step definition under src/test/java as StepDefinition. Create a class called CucumberReport.java inside StepDefinition package. import io.cucumber.java.en.*; public class CucumberReport { @Given("User is on Welcome Page") public void user_is_on_welcome_page() { System.out.println("User on welcome page"); } @When("User clicks on Login button") public void user_clicks_on_login_button() { System.out.println("User clicks on Login button"); } @Then("Welcome page should be displayed") public void welcome_page_should_be_displayed() { System.out.println("Welcome page should be displayed"); } } Step 6: Create a runner class as TestRunner.java. import org.junit.runner.RunWith; import io.cucumber.junit.Cucumber; import io.cucumber.junit.CucumberOptions; @CucumberOptions(features = ".\\src\\test\\resources\\Feature\\CucumberReport.feature", plugin = {"pretty", "json:target/cucumber-reports/reports.json", "junit:target/cucumber-reports/Cucumber.xml", "html:target/cucumber-reports/reports2.html"}, monochrome = true) public class TestRunner { Step 7: Click right click on TestRunner.java file to choose Run As -> JUnit Test. After successful build, three types of reports will be generated - HTML report, XML report, Json report. Among all three types of reports, HTML reports are the most readable. In plugin parameter, we have to mention the target folder and the destination folder for the reports to be generated. When we open reports2.html by double clicking, it opens in browser and gives us the clear picture about the feature file implemented. It is very easy to generate inbuilt HTML reports for Cucumber as we don't need to add any dependency in pom.xml but we can explore many other features for a report. In the next blog, I will discuss about Extent reports which gives better overview about our implementation. Cucumber provides a built-in feature to generate reports, allowing developers to create detailed test execution summaries and improve the visibility of test results. This tutorial will explore how to generate reports with Cucumber and leverage the available features to enhance test reporting capabilities. #### Choosing a Formatter Cucumber offers various formatters to generate reports in different formats. Developers should choose a formatter that suits their reporting requirements, such as HTML, JSON, JUnit, or pretty. #### Specifying the Output File Developers must specify the output file path and name where they want to save the generated report. The `--out` flag followed by the file path is used for this purpose. #### Running Cucumber with the Formatter To generate a report, developers should execute their Cucumber tests with the chosen formatter by providing the `--format` flag followed by the formatter name and appending the `--out` flag with the output file path. #### Viewing the Generated Report After test execution completes, developers can open the generated report file in a web browser or use appropriate tools to view the report. The report displays a summary of the test execution along with detailed information about each scenario and step. Common mistakes include not specifying the output file path correctly, using an inappropriate formatter, forgetting to include the formatter option when running Cucumber, and frequently asking questions such as customizing the generated report layout, generating reports in multiple formats simultaneously, using plugins to enhance reporting capabilities, or including screenshots or attachments. Integrating files with your report is a breeze! Just refer to the chosen formatter's documentation for tips on capturing extra details. Now, about combining Cucumber reports with Continuous Integration tools like Jenkins or Bamboo: you can either publish generated report files as artifacts or use specific plugins that seamlessly integrate with Cucumber reports.